

服务描述

对提供的一张人脸图像进行基于人脸的属性检测，返回对应的表情信息，包括惊讶、恐惧、厌恶、喜悦、悲伤、愤怒、中立，将检测结果以 Json 格式返回给调用者。

接口调用

本节将对人脸图像表情识别服务进行阐述。由于实际所使用到的编程语言较多，此文档不会一一涉及。本章中将只就 Java 语言如何调用识别服务接口进行代码示例讲解。其它语言如何进行调用，用户可参照接口调用例程。

HTTP 请求公共部分说明

HTTP 头

| Key | Value | 请求或响应 |
|-----------------------|------------------|---------------------------------------|
| Content-type | application/json | 存在于请求和响应中，不可为空 |
| Abis-Request-Sequence | 请求流水号 | 存在于请求和响应中，请求中可以为空 如为空则由服务自动生成一个流水号 |

服务调用状态返回码

返回的错误码是错误类型，具体错误信息请查看返回 content 中的内容

| 错误码 | 说明 | 解决方案 |
|-------|----------|--|
| 2000 | 成功 | |
| -2001 | 未知的异常 | 请查看返回的 content 中内容 |
| -2020 | 传入图片内容为空 | 请确定是否传入了图片 |
| -2021 | 加载图片失败 | 传入的不是有效图片或不支持的图片格式，支持的图片格式包括： BMP/JPG/TIF/PNG/WSQ/JP2/JPC/J2K/TFF |

| | | |
|-------|----------------|----------------------------------|
| -2100 | 未检测到人脸 | 请确保图片是正面朝上的人脸图片，且人脸清晰，图片背景没有反光 |
| -2101 | 检测人脸属性异常 | 详情请查看后台日志 |
| -2102 | 接口调用超时 | 可能是请求量过大，或者可能是服务器上 cpu 被其他应用占用过大 |
| -2108 | 文件读取失败 | 请查看 content 字段返回的异常信息 |
| -2109 | 算法调用图像质量检测接口失败 | 详情请查看后台日志 |
| -2119 | 人脸属性检测的条件参数未设置 | 请输入需要检测的人脸属性参数 |

HTTP 方法及 URL

HTTP 方法构建

HTTP 方法，要求必须为 Post 方法；

URL 构建：<http://IP:Port/feeling/v1/detect/attribute>

URL 举例：<http://abis.cn:8080/feeling/v1/detect/attribute>

Post 实体内容

Post 实体内容仅为一个 Json 对象，说明如下：

1) 实体内容定义

| 必填 | JSON Path | 类型 | 含义 | 注意事项 |
|----|-------------------------------------|---------|----------------|---|
| 是 | \$.image.content | String | Base64 编码的图像数据 | 需要传入人脸图像 |
| 是 | \$.faceImageAttributeDetectionOptio | Object | 检测属性开关 | 根据业务需要设置检测项，开启的检测项越多，耗时越多，请按需设置 |
| 否 | \$.rotateEnable | Boolean | 是否自动旋转人脸方向 | 如果图片中人脸不能保证是朝上的，可设为 true。开启自动旋转会增加耗时。 自动旋转相关说明详见“注意事项” |

其中，faceImageAttributeDetectionOptio 定义如下：

```
boolean detectFeeling; //是否检测表情
```



```
        "surprise": 0,
        "fear": 0,
        "hate": 26,
        "happy": 4,
        "sad": 7,
        "anger": 4,
        "neutrality": 56
    }
}
],
"response": {
    "status": 200,
    "metadata": {},
    "content": ""
}
}
```

示例代码



Checking demo_JS.zip



Face_checkingDemo_java.zip

注意事项

1) 对图像要求

支持的图片格式：BMP/JPG/TIF/PNG/WSQ/JP2/JPC/J2K/TFF

对比度、亮度适中；

尽量不要有较多背景，尽量选择浅色、纯色背景；

图像尽量避免反光和黑影；

图像清晰，避免图像模糊不清； 图像分辨率不宜过高，一方面造成图片过大，不利于传输，一方面分辨率过高也会影响识别率。

人脸图片中，人脸宽度不低于 60 像素

2) 服务需要联网调用

由于当前服务基于 http 技术实现，用户调用服务前请确保您的设备可以连通服务器，并保证一定的带宽。否则图像数据网络传输较慢，会引起较长时间等待。

3) 若接口涉及到数据需要进行 base64 编码，请引用 commons-codec-1.13.jar 对图片进行 base64 编码：`base64.encodeBase64String(byteData)`;

4) 关于自动旋转功能的说明

非正向人脸图片无法直接检测到人脸属性。为了解决部分图片中人脸向下或向侧的问题，增加图片的自动旋转功能。对于待检测图片中人脸方向非正向的情况，可以自动旋转为正向并检测相关属性。自动旋转功能可以开启或关闭，在入参中控制。

对于多人脸情况，以方向相同的人脸数最多的为旋转方向，如果图片中有三个人脸，其中一个朝左，两个朝下，取人脸朝下的方向进行旋转，旋转后两个人脸向上，可检测两个人脸的属性。

调用参数说明：

1) 【入参】是否检测并旋转图片方向 `rotateEnable`

`true` 表示开启图片自动旋转，影响范围：返回值中 `faceID`、属性信息

`faceParam[0].xxx` 都是基于旋转到正向的图片；

`false` 表示关闭图片自动旋转，影响范围：对于非正向的人脸，不进行旋转，返回的人脸属性请参考以下说明

2) 【入参】是否检测旋转坐标 `detectFpcn`

true 表示检测，影响范围：返回值中 faceParam[0].fpcn 显示旋转前人脸框角度及坐标；

false 表示不检测，影响范围：返回值中 faceParam[0].fpcn 为空

3) 【返回值】图片 ID faceParam.faceId

如果人脸正向，正常返回 faceID

如果人脸非正向，且 rotateEnable =true，正常返回 faceID

如果人脸非正向，且 rotateEnable =false，且 detectFpcn=true，返回 faceID=-1

如果人脸非正向，且 rotateEnable =false，且 detectFpcn=false，返回的人脸属性是完全基于算法返回的结果，可能检测不到人脸，返回为空，也可能检测到了人脸，但是返回的人脸属性是不可用的；

4) 【返回值】人脸矩形框 faceParam[0].rectangle

如果人脸正向，返回正常坐标

如果人脸非正向，且 rotateEnable =true，返回旋转为正向后人脸的坐标

如果人脸非正向，且 rotateEnable =false，且 detectFpcn=true，返回

rectangle= (0, 0, 0, 0)

如果人脸非正向，且 rotateEnable =false，且 detectFpcn=false，返回的人脸属性是完全基于算法返回的结果，可能检测不到人脸，返回为空，也可能检测到了人脸，但是返回的人脸属性是不可用的；

5) 【返回值】人脸矩形框旋转相关数据 faceParam[0].fpcn (含旋转角度、人脸 xy 坐标及宽高)

如果 detectFpcn=true，返回原始图中人脸旋转角度 (0 或 90 或 180 或-90) 及人脸框坐标

如果 detectFpcn=false，返回空”

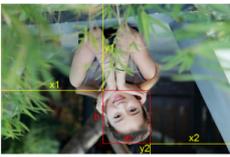
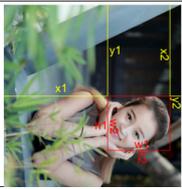
场景应用建议：

如果目标图片基本都是人脸正向，个别非正向的图片允许检测不到属性，建议传参 rotateEnable =false, detectFpcn=false，以便提高响应速度

如果目标图片有可能人脸非正向，希望非正向人脸也检测到属性，建议传参 rotateEnable = true;

如果目标图片有可能人脸非正向，希望检测到人脸框在原始图中的坐标以及人脸角度（以便用于人脸抠图等业务需要），建议传参 `detectFpcn=true`。

图片示例：

| 目标图片 | 入参 | | 返回值 | | | |
|--|-------------------------------------|-------------------------------------|-------------------------------|--|---|--|
| | 是否自动旋转 <code>rotateEnable</code> | 是否检测旋转坐标 <code>detectFpcn</code> | 返回图片ID <code>faceID</code> | 返回人脸属性 <code>faceParam[0].xxx</code> (如性别、表情等) | 返回人脸矩形框坐标 <code>faceParam[0].rectangle</code> (旋转到正向后的坐标) | 返回人脸框旋转角度及坐标 <code>faceParam[0].fpcn</code> (原始坐标) |
|  <p>人脸向上</p> | <code>true</code> | <code>true</code> | 有效ID | 正常 | <code>(x,y,w,h)</code> | <code>(0°, x,y,w,h)</code> |
| | <code>true</code> | <code>false</code> | 有效ID | 正常 | <code>(x,y,w,h)</code> | 空 |
| | <code>false</code> | <code>true</code> | 有效ID | 正常 | <code>(x,y,w,h)</code> | <code>(0°, x,y,w,h)</code> |
| | <code>false</code> | <code>false</code> | 有效ID | 正常 | <code>(x,y,w,h)</code> | 空 |
|  <p>人脸向下</p> | <code>true</code> | <code>true</code> | 有效ID | 正常 | <code>(x2,y2,w,h)</code> | <code>(180°, x1,y1,w,h)</code> |
| | <code>true</code> | <code>false</code> | 有效ID | 正常 | <code>(x2,y2,w,h)</code> | 空 |
| | <code>false</code> | <code>true</code> | -1 | 空 | <code>(0,0,0,0)</code> | <code>(180°, x1,y1,w,h)</code> |
| | <code>false</code> | <code>false</code> | 空或-1 | 空或无效的值 | 空或无效的值 | 空或无效的值 |
|  <p>人脸向侧</p> | <code>true</code> | <code>true</code> | 有效ID | 正常 | <code>(x2,y2,w2,h2)</code> | <code>(90°, x1,y1,w1,h1)</code> |
| | <code>true</code> | <code>false</code> | 有效ID | 正常 | <code>(x2,y2,w2,h2)</code> | 空 |
| | <code>false</code> | <code>true</code> | -1 | 空 | <code>(0,0,0,0)</code> | <code>(90°, x1,y1,w1,h1)</code> |
| | <code>false</code> | <code>false</code> | 空或-1 | 空或无效的值 | 空或无效的值 | 空或无效的值 |